

---

## 3 Wettbewerb: Seitenverbrauch minimieren

In der Aufgabe #3 geht es darum Nachrichtenaustausch und erweitertes Paging in StuBSMI einzubauen. Im Laufe der Implementierung neigen viele dazu das wiederverwenden von Seiten zu vernachlässigen: Seiten werden nicht mehr ordentlich frei gegeben, wenn sie nicht mehr gebraucht werden.

Da dies ein unhaltbarer Zustand für ein ordentliches Betriebssystem ist und wir selbstverständlich ein ordentliches Betriebssystem bauen wollen, müssen wir uns dringend darum kümmern. Daher gibt es diesen Wettbewerb der Implementierungen.

Auf eurer StuBSMI Implementierung soll eine von uns bereitgestellte Anwendung laufen. Diese Anwendung wird sich an die von uns vorgeschlagene API halten.

### 3.1 Die Anwendung

```
#include "syscall.h"

char __attribute__((aligned(4096))) sbuf[4097];
char __attribute__((aligned(4096))) rbuf[4097];

void main() {
    fork();
    fork();
    int ppid = getpid();
    int other = fork();
    if (ppid == other) {
        /* child */
        sbuf[0] = 3;
        sbuf[4096] = ppid;
        send(other, sbuf, 4097, rbuf, 4097);

        char msg[] = "REPLY:␣AA\n";
        msg[7] += rbuf[0];
        msg[8] += 3 + ppid;
        write(/*fd */ 0, msg, /*len=*/ 10, /* x= */ -1, /*y*/ -1);
    } else {
        /* parent */
        int X = recv(rbuf, sizeof(rbuf));
        rbuf[0] = rbuf[0] + rbuf[4096];
        reply(X, rbuf, 4097);
    }

    exit();
}
```

Die Anwendung wird auch unter `/proj/i4bs/vorgaben/wettbewerb.cc` verfügbar sein. Diese Anwendung spaltet sich mit `fork()` in 4 Paare von Fäden auf. Innerhalb jedes Paares wird ein Faden auf eine Nachricht warten (parent) und der andere Faden (child) schickt seinem Partner eine Nachricht. Der empfangende Faden errechnet aus der Nachricht einen Wert und sendet den Buffer zurück. Am Ende beenden sich alle acht Fäden mit `exit()`. Den Systemcall `exit()` muss dabei noch implementiert werden. Dies ist auch der Ort an dem Seitenkacheln freigegeben werden können.

Als Ausgabe wird das Programm etwas ähnliches ausgeben wie:

```
REPLY: DD
REPLY: GG
REPLY: EE
REPLY: FF
```

Die Zeilen können in beliebiger Reihenfolge auftreten. Die beiden Buchstaben nach dem `REPLY` hängen dabei von eurer Implementierung der Fadenummer ab. Allerdings müssen die beiden Buchstaben, die folgen, in allen Fällen gleich sein!

### 3.2 Die Messung

Beim Wettbewerb geht es zum einen darum nicht mehr benötigte Seiten wieder ordentlich frei zu geben (**Vorraussetzung!**). Und möglichst wenig Seiten zu verbrauchen. Um dies vergleichbar zu machen müsst ihr die Anzahl der freien Seiten messen, die euer Allokator verwaltet. Diese Zahlen müsst ihr an zwei Stellen ausgeben:

1. Direkt vor dem `scheduler.schedule()` in der `main()`
2. Am Anfang der `IdleThread::action()`

Dabei sollt ihr die Anzahl der verfügbaren Kernelpages und die Anzahl der verfügbaren Userpages ausgeben. Die Anzahl der freien Seiten darf dabei steigen, aber auf keinen Fall fallen! Und ihr dürft keine Seiten ausgeben, die in den Multibootinformationen *reserved* sind.

---

Der QEMU soll mit 128 MiB (default) RAM ausgestattet sein.

### 3.3 Was sagt die Musterlösung?

Die Musterlösung ignoriert alle Seiten unterhalb von 0x100000. Unser Speichererkennungsmechanismus erkennt 32478 freie Seiten. Die Regionen die dafür bei uns verwendet werden sind folgende:

```
module 0x0x118000-0x0x11f000 user/build/initrd.img
kernel 0x100000-0x116010
(available) 0x100000 0x7ffdfff
 \-> use! 0x100000-0xfffff 0 pages
 \-> use! 0x117000-0x117fff 1 pages
 \-> use! 0x121000-0x7ffdfff 32477 pages
(reserved) 0x7ffe000 0x7ffffff
(reserved) 0xfffc0000 0xffffffff
```

Am ersten Messpunkt benötigt unser System 49 Seiten:

```
allocator: 49 pages used
allocator: 28669 free userpages; 111 MByte
allocator: 3760 free kernelpages; 14 MByte
```

Am zweiten Messpunkt benötigt unser System 42 Seiten:

```
allocator: 42 pages used
allocator: 28670 free userpages; 111 MByte
allocator: 3766 free kernelpages; 14 MByte
```

Diese Abnahme im Seitenverbrauch liegt daran, dass im ersten Messpunkt sowohl ein Benutzerfaden als auch der Idlefaden existieren. Am zweiten Messpunkt existiert nur noch der Idlefaden. Unsere Initial Ramdisk verbraucht 7 Seiten.

### 3.4 Der Preis

Ruhm. Ehre. Unsterblichkeit. Gummibärchen.

**Abgabe: am 15.07.2015**